How to not break LTE crypto

Benoit Michau and Christophe Devine
benoit.michau@ssi.gouv.fr
christophe.devine@ssi.gouv.fr

ANSSI

Abstract. The LTE standard defines a strong security model and architecture for protecting 4G mobile communications. However, it is yet very unclear how the various modems available on the market are implementing and enforcing the LTE security procedures.
In this paper, we first introduce the basics of LTE security. Then, we show multiple LTE security bypasses that we found in the different 4G modem implementations we tested. We also describe two issues we found in WCDMA stacks from our previous research. Finally, we dive into a few 4G modem implementations to see how they are interworking with the Android OS, and how one can try to get information out of them. To conclude, we propose improvements, on the terminal side but also on the network side, in order to increase the effective security level of 4G communications.

Due to a request from Mediatek, a part of the original article has been redacted in section 3.

Acknowledgement. We would like to thank José Lopes Esteves, Thomas Cordier, Jérôme Plût, Loïc Habermacher, more widely the whole wireless security lab of ANSSI, the Network and Security Assessment team from Orange, the people at Sysmocom and Amarisoft for providing great products and finally the security teams at Qualcomm and Samsung.

1 LTE security introduction

Please note:
For the reader not familiar with cellular-specific terms and acronyms, definitions are provided for each of the abbreviations in section 5. This article is not intended to provide either a detailed description of the architecture of cellular networks, or details on the security procedures and algorithms. The article published in french at SSTIC 2014 [9] already provides a general description of 2G, 3G and LTE mobile networks. It is also easy to find good resources in english on the Internet, for example from the most generic to more detailed ones, on Wikipedia [12], radio-electronics [13], or sharetechnote [24]; the most accurate descriptions being found in the 3GPP standards [2] themselves.

1.1 Introducing the LTE network

The LTE architecture and its security model were initially defined in 3GPP Release 8, in 2008. The security aspects of LTE are detailed in the specification TS 33.401 [6]. Since Release 8, new features have been developped, some of which have major impacts on the LTE security model. However, in the context of this study, we mainly focus on the basics of LTE security as defined in its initial release.

In this article, we will always refer to the LTE network as the global 4G mobile network technology. In the standards' terminology however, *LTE* sometimes refers to the *radio access network* (the base stations), whereas *EPC* refers to the *core network* (abbreviated CN) part only, and EPS to the whole mobile network (LTE and EPC).

1.2 Basic principles of an LTE network

The radio access network is composed of all the LTE base stations, called *eNodeB*, that provide the radio coverage across the whole country. For the metropolitean France area, the ANFR agency reported in a survey in november 2015 [18] a total of 22.110 4G radio sites, for all the four french MNOs. Each radio site hosts one or multiple eNodeBs; each eNodeB runs one to several cells. Every MNO splits all the covered areas into tracking areas, each of which groups several eNodeBs under a given *tracking area code* (abbreviated TAC). The general architecture and principles describing the LTE radio access network are given in the 3GPP specifications TS 36.300 [7] and TS 36.401 [8].

The core network is composed of fewer equipments, which are responsible for managing the mobility and sessions of all LTE terminals across all the eNodeBs, and for routing their data trafic towards IP networks (Internet, VoIP platforms, ...). Those equipments are:

- MME, responsible for handling all the signaling towards eNodeB and LTE terminals, including mobility and session management;
- SGW-PGW, responsible for routing user data (actually IP packets) to and from 4G terminals;
- HSS, responsible for storing and delivering subscribers information and authentication data.

Each MNO has datacenters where MME, SGW-PGW and HSS are installed. The specification TS 23.401 [4] describes the main architecture for the LTE core network.

All the eNodeBs are connected to the core network through an interface named S1; eNodeBs can also be connected together through an interface named X2, in order to smooth mobility from one cell to another. The colourful figure 1 illustrates the basic architecture of an LTE network.

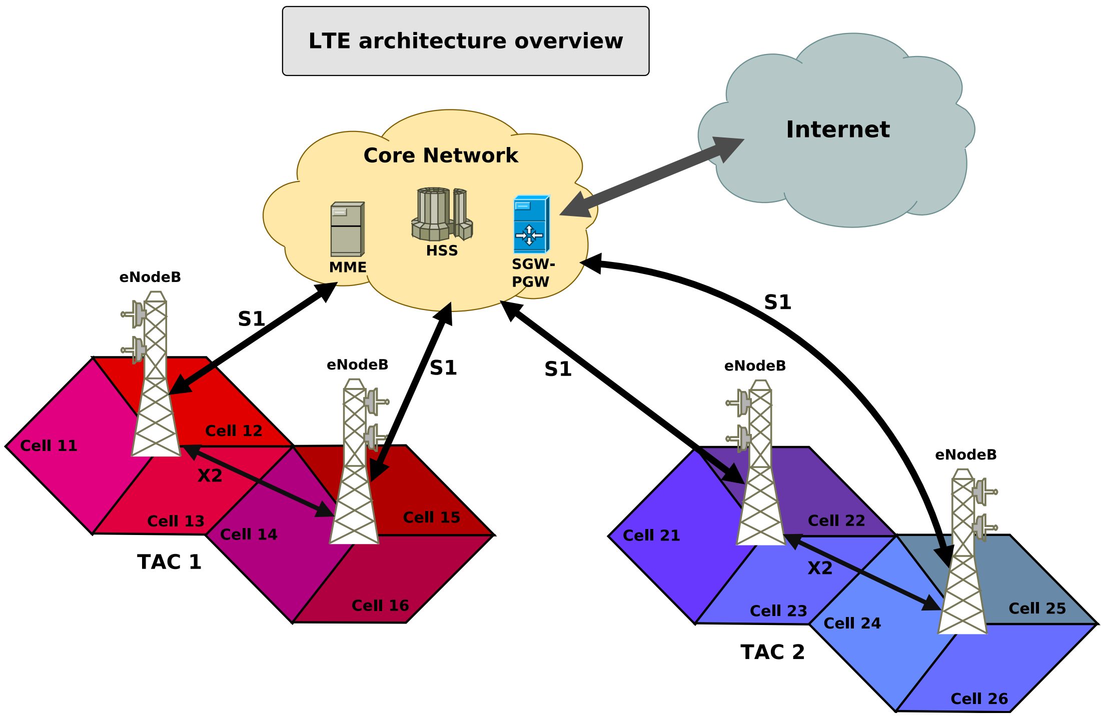

Fig. 1. Basic architecture of an LTE network

When a terminal, actually called *UE* (User Equipment) in 3GPP terminology, connects to the Internet through an LTE network, the following sequence of actions takes place:

- The terminal establishes an initial bi-directional radio channel with an eNodeB for transporting signaling only; within this *signaling radio bearer* (abbreviated SRB), the RRC protocol is used between the terminal and the eNodeB in order to further configure the radio channels;
- The terminal then connects to an MME in the core network through the S1 interface of the eNodeB; the NAS protocol is used between the terminal and the MME in order to manage mobility (including authentication and master key establishment) and data session establishment;
- After the terminal fulfilled all the security procedures requested by the MME and provided all the required data session parameters, the

MME requests the eNodeB to modify the previously established radio channel in order to convey user data;

- The eNodeB uses RRC signaling with the terminal to establish a *data radio bearer* (abbreviated DRB);
- The terminal can send and receive user data within this DRB; this user data is forwarded by the eNodeB to the a SGW-PGW in the core network and to the Internet.

Figure 2 shows the signaling stacks over the radio interface between a terminal and an eNodeB, and over the S1 interface between an eNodeB and an MME; it also indicates the references to the standards that describe the inner workings of each part of the stacks.

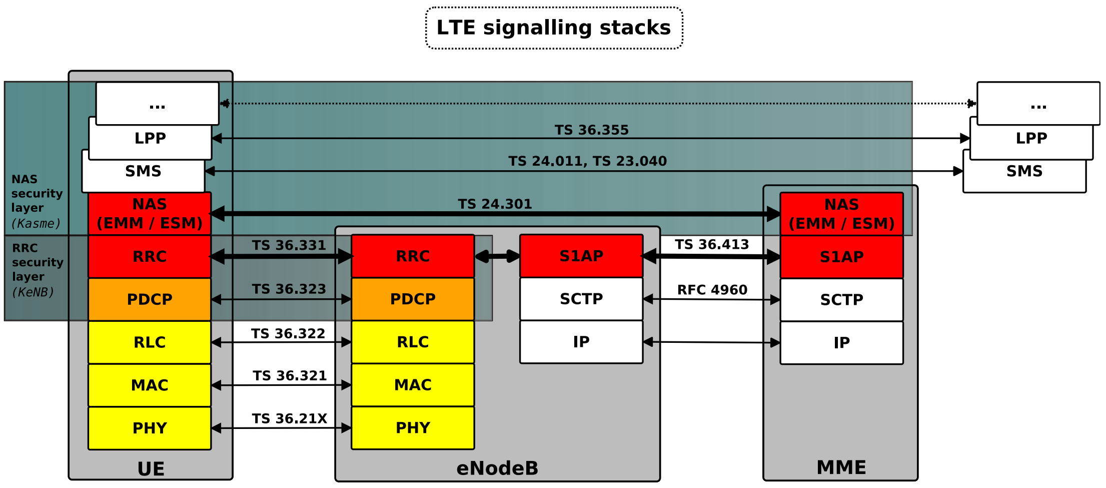

Fig. 2. Signaling stacks within an LTE network

1.3 Basic principles of LTE security

The LTE standard reuses many security concepts developed for UMTS networks:

- The USIM is reused to manage mutual authentication between the home network of the subscriber and the subscriber itself; the authentication key is still 128 bit long and the Milenage algorithm makes use of it;
- User data and radio signaling are security protected (ciphered for user data, ciphered and integrity protected for signaling) at the PDCP layer, between the terminal and the eNodeB;

- Ciphering is optional, whereas integrity protection for signaling is mandated, excepted for unauthenticated emergency sessions;
- The SNOW 3G cryptographic algorithm is reused for encryption and integrity protection;
- The principle of selecting a given cryptographic algorithm, through the *security mode control* procedure (abbreviated SMC), is kept;
- Temporary identities are delivered and renewed within a secured signaling channel, so that terminals do not send their permanent identity (IMSI) in clear every time.

However, additional developments have been added:

- The authentication algorithm has been extended to ensure a strict cryptographic separation between LTE and older networks;
- Cryptographic material established during the authentication stage is further derived, taking as input parameters the mobile country code and mobile network code (MCC and MNC) of the core network to which the terminal is attaching (which is different than the subscriber's home network in case of roaming);
- A 2nd level of security has been added at the NAS layer (with ciphering and integrity protection), in order to secure the end-to-end link between the terminal and the MME;
- AES has been added as a cryptographic algorithm for encryption and integrity protection, followed by ZUC (a chinese cryptographic algorithm) in the 11th release of the standard;
- The security protection of specific privacy-related signaling (such as IMEI, data-session parameters, geo-positionning) is explicitely mandated for the RRC and NAS protocols.

1.4 Authentication and key establishment

The 3G authentication protocol (i.e. 3G-AKA) is reused as it has proven its efficiency since its initial development in 1998. However, the keys produced during its execution {CK, IK} are not directly used for protecting signaling or user data in LTE. They are derived further into a new master key, called *Kasme*, using the MCC and MNC of the core network to which the terminal is attaching as input parameters. Therefore, an LTE authentication vector provided by an HSS to an MME is the set {*RAND, AUTN, XRES, Kasme*}. This has two advantages:

- *Kasme* is outputted by an HMAC-SHA256 function, and is hence 256 bit long, enabling the potential use of 256 bit ciphering and integrity protection keys in a future release of the LTE specification.

- *Kasme* is derived using MCC and MNC, which leads to the distribution of authentication vectors specifically built for each given roaming partner. This makes it possible for every MNO to enforce the security of authentication vectors delivery through their roaming interfaces. For instance, if this enforcement is correctly implemented by french MNOs, an MNO from abroad will not be able to obtain authentication vectors for a french LTE network (with MCC 208), and so will not be able to fake a french LTE base station.

A marker has been added in the authentication parameter *AMF* (meaning *Authentication Management Field*) because of this evolution of the key derivation scheme. *AMF* is part of *AUTN* and delivered to the USIM inside the terminal when the MME sends an authentication request to it. A specified bit set to 1 in this *AMF* field enables the terminal to distinguish an authentication vector produced by its HLR for a connection to a 2G or 3G network, from a vector produced by its HSS for a connection to an LTE network. When a terminal connects to an LTE network, it must ensure any vectors received within authentication requests have this AMF bit set to 1. If this is not the case, the terminal must disconnect from the LTE network. This bit within the AMF field is crucial for the security of LTE connections:

- It differentiates 2G/3G and LTE authentication vectors, where 2G/3G vectors can be obtained easily from almost any SS7 connection point worldwide, contrary to LTE vectors when the subscriber's home operator enforce the security of authentication vectors delivery (as explained above);
- Since the *AMF* field is taken into account to compute the MAC part of *AUTN*, which is verified by the USIM in the terminal, the subscriber is sure the LTE network it connects is specifically allowed by his home operator;
- It also allows the terminal to enforce the use of distinct authentication keys and/or algorithms for 2G/3G and for LTE by the USIM, in case the subscriber's mobile operator has decided to do so (this is not a common practice, to our knowledge);
- It enables in a future release of the LTE specification to extend the key derivation scheme for LTE in the USIM.

1.5 Algorithm negotiation and security activation

As soon as the authentication between a terminal and an MME is successful, the security can be activated at the NAS layer. Similarly to the procedure in 3G, a *security mode control* procedure is run to select cryptographic algorithms and start the integrity protection and potential encryption of the NAS signaling.

During its initial attachment, before the NAS link is secured, the terminal reports its capabilities, including its security capabilities (i.e. which cryptographic algorithms are supported by the baseband). Then, the security mode control procedure is initiated by the MME, which selects which algorithm to use for encryption:

- EEA0 (no encryption);
- EEA1 (SNOW 3G);
- EEA2 (AES in counter mode);
- EEA3 (ZUC);

and for integrity protection:

- EIA1 (SNOW 3G);
- EIA2 (AES in CBC-MAC mode);
- EIA3 (ZUC).

The MME replays the security capabilities that were received from the terminal beforehand, and finally applies the integrity protection but not the encryption. When this NAS message is received, the terminal checks which algorithms are selected by the MME, verifies the integrity of the message and finally verifies that the replayed security capabilities have not been tampered with. This is to ensure there is no man-in-the-middle attacker who tries to downgrade the security level of the connection.

If everything is successfully verified by the terminal, it then responds with a security mode complete NAS message to the MME, which is both encrypted and integrity protected. After this procedure is completed, the NAS protocol layer between the terminal and the MME is secured.

From here, the terminal can request the establishment of a data radio bearer to transport user data. However before this can happen, the MME has to further derive *Kasme* into a key for the eNodeB to which the terminal is currently connected, namely *KeNB*. As soon as the eNodeB receives *KeNB* and the UE security capabilities from the MME, it uses them to secure the radio channel:

- The eNodeB chooses which cryptographic algorithms to use for encryption of user data, of the RRC signaling, and for the integrity protection of the RRC signaling;

- The eNodeB starts a security mode control procedure towards the terminal at the RRC level; this procedure is similar in its principle to the one at the NAS level, except UE security capabilities are not replayed again;
- After a successful verification by the terminal and the completion of the procedure, a key set derived from *KeNB* is used to secure all user data (in the DRB) and RRC signaling (in the SRB) between the terminal and the eNodeB.

A terminal does not stay connected to an eNodeB all the time, but only when sufficient user data is buffered and has to be transferred. When a terminal reestablishes the radio connection with an eNodeB to transfer user data, a *key set identifier* (abbreviated KSI) is used between the terminal and the MME to indicate which *Kasme* to take into account (up to seven LTE key sets can live in parallel for a given terminal). The MME then derives a fresh *KeNB* to be used by the eNodeB and the terminal during a new RRC *security mode control* procedure. From this point again, all user data in the DRB and RRC signaling in the SRB are secured.

1.6 Further information

Only the most fundamental security procedures for the connection of terminals to an LTE network have been described here. There are more procedures and additional complexity related to the mobility within an LTE network, and between LTE and 2G/3G networks. However, for this study, we only focus on those basic procedures.

In order to go deeper into the understanding of the LTE security architecture, the resources cited in the introduction of this section can be consulted. For the french reader, an introduction to the LTE security was presented at the CAESAR conference in 2011: «La sécurité dans les réseaux mobiles LTE» [15].

2 LTE modem vulnerabilities

2.1 Introduction to the LTE testing process

Because LTE technology has spread all over the world and is becoming ubiquitous, we believe it is essential to test the terminal equipments that implement it. We have developped an LTE test-bed which is security oriented and allows us to evaluate the implementation of a part of the LTE security procedures. This is not an easy task, hopefully more and more open-source projects and affordable products will become available and ease this testing process.

In this section, manufacturers are anonymized in order to avoid pin-pointing any of them specifically. We could find, for each of the LTE modem we tested, at least one of the vulnerabilities described in this article.

We explain in this section how it is possible to bypass certain parts of the LTE security procedures in order to obtain private information from terminals and LTE subscribers. These bypasses may enable the interception of all user data in clear over the LTE radio interface, or may simply allow getting private signaling information (such as IMEI, data session parameters, ...). All of them can be triggered by running an illegitimate LTE eNodeB. The principle of the fake LTE base station is illustrated in figure 3 and is really simple: an attacker runs an eNodeB which broadcasts the same network identifiers as a legitimate network (e.g. MCC 208, MNC 30 in the figure). This is always possible as broadcast control trafic is never authenticated in mobile networks. On the other side, the attacker has neither access to the authentication key, nor to any intermediate security contexts of the subscriber.

In these conditions, a subscriber who is close enough to the attacker's eNodeB will camp on it: his LTE terminal will lock on the eNodeB synchronization signal, connect to it and try to access the Internet. If all the LTE security procedures are well implemented in the terminal, nothing except the IMSI of the subscriber will be obtained by the attacker, and the terminal will release the connection, switching back to a legitimate eNodeB after detecting a security violation in the NAS or RRC signaling. However, we will see below that many bugs and bypasses can be found in LTE modem stacks in order to access more than the victim's IMSI.

A major point here is that all these bypasses do not require breaking any of the cryptographic algorithms as standardized by the 3GPP. They are only due to failures from modem manufacturers to correctly implement all the security checks required by the standard.

2.2 Previous work

Here we detail important work that have been published in the last years regarding security issues found in LTE terminal and subscriber's equipments.

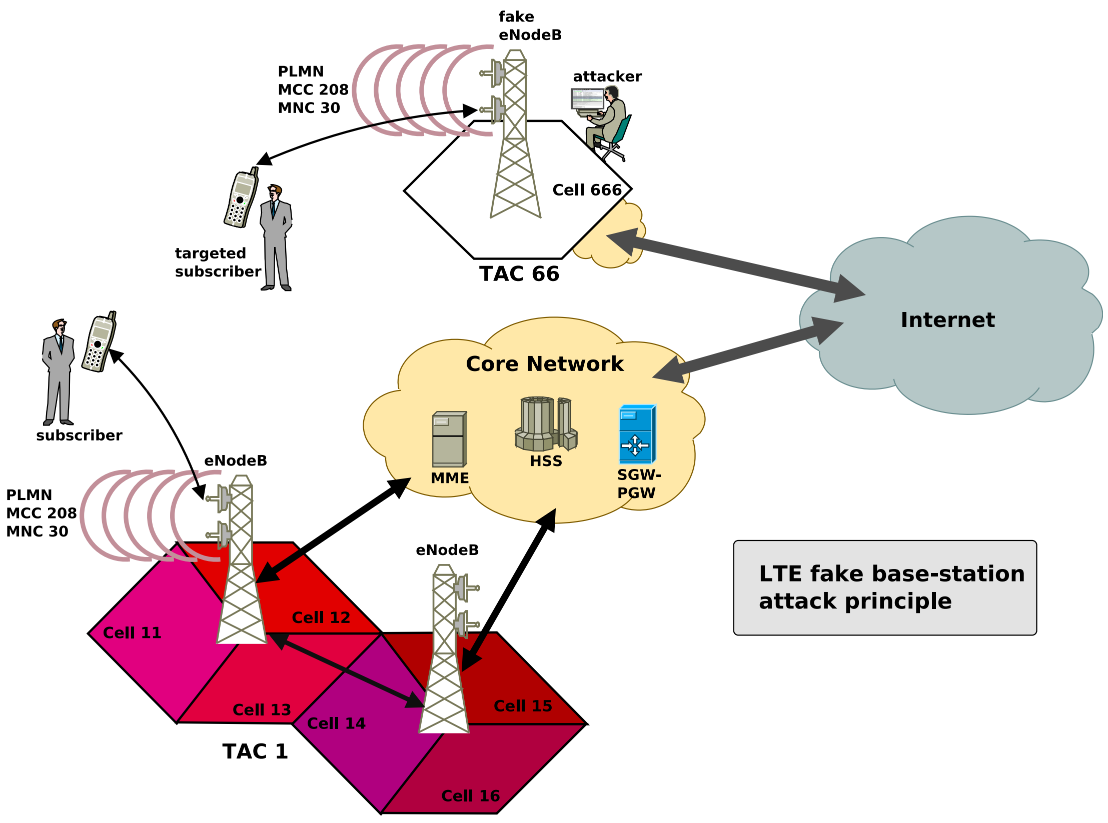

Fig. 3. Principle of the LTE fake base-station attack

EIA0, no integrity protection of the signaling

EIA0 is the null algorithm for integrity protection of the signaling. It is defined in the 3GPP standard, and must only be used for unauthenticated emergency sessions.

However, we identified in [9], that it was possible to trigger the selection of EIA0 at the NAS and RRC layers towards a targeted LTE terminal for any kind of LTE connection. This was exploitable against all modems from a specific manufacturer up to the end of 2012; a patch was deployed in early 2013.

Exploitation of this issue allows a complete LTE connection interception with a fake LTE base station. In case the legitimate network also supports this insecure EIA0 mode (in its eNodeBs and MMEs), this enables a full man-in-the-middle attack on the LTE radio interface. For this attack to succeed, the attacker needs only to guess the security capabilities of the terminal, and the KSI in use between the victim's terminal and the legitimate network. With this knowledge, the attacker can run successful *security mode control* procedures at the NAS and RRC layers, intercepting all the user's trafic and signaling in clear.

Authentication key extraction from USIM cards

In all cellular technologies (2G, 3G and LTE), every session key is derived from the unique authentication key which is stored in the USIM of each subscriber, and symetrically in the HLR / HSS of the MNO. The possibility of extracting an authentication key from an USIM as explained in the article «Small Tweaks do Not Help: Differential Power Analysis of MILENAGE Implementations in 3G/4G USIM Cards» [25] enables for passive decryption of all past, present and future cellular communications of the subscriber.

This is because there is no *Perfect Forward Secrecy* in mobile networks key establishment, including within LTE networks. This powerful attack however requires physical access to the USIM of the victim and the knowledge (or the bypass of the verification) of its PIN code.

LTE subscriber location tracking

In the article «Practical attacks against privacy and availability in 4G/LTE mobile communication systems» [22], which was also introduced at the Black Hat EU 2015 conference, the authors found that it is possible to obtain *UE measurement report*, *UE information response* and *radio link failure* reports from all LTE modems, without the need to enable the security of the RRC protocol. This makes it possible for an attacker to get this information from surrounding terminals with a fake LTE base station and to deduce localization information for all modems caught.

Persistent targeted denial of service

In [9], we identified a bug which made it possible to send an out-of-sequence *Authentication Reject* NAS message to a terminal, leading to its definitive deconnection from the network. This caused the terminal to enter the specific state *EU3: ROAMING NOT ALLOWED*, as defined in TS 24.301 [4], in which the terminal sets its home PLMN in its list of forbidden networks...

In [22], similar cases were identified with standard mobility management NAS messages: *Attach Reject*, *Tracking Area Update Reject* and *Service Reject*, containing reject causes that lead to the same *EU3* state. They also identified the possibility of denying the service for LTE only, hence downgrading the terminal to the 2G / 3G network.

2.3 LTE testing infrastructure

We have described several vulnerabilities that were previously identified during our research. Before presenting the new vulnerabilities that were found, we'll describe briefly our testing environment :

All the tests we are conducting are run within a Faraday cage, with our own network equipments, handsets and USIM cards. In this way, we are completely isolated from the surrounding mobile networks and handsets, and do not risk interfering with them. We use a standard eNodeB from Amarisoft (software-defined, running on an x86 PC with an USRP serie N), together with a custom-made tiny LTE core network. Our LTE core network is entirely written in Python, includes an ASN.1 compiler and a PER codec, and is available on github: corenet [17].

Other open-source projects allow experimenting with LTE networks and can be used in order to build an LTE testing infrastructure, most notably:

- OpenAir Interface from EURECOM [19]: the most complete open-source project related to LTE; it provides the implementation of an LTE eNodeB, core network and UE, and supports USRP and BladeRF radio peripherals.
- OpenLTE [20]: an integrated single cell LTE network, running with an USRP serie B radio peripheral.
- srs-lte and srs-ue [21]: a library which implements LTE radio processing and a terminal (User Equipment) application.

For the issues discovered below, we have worked with several smart-phones with LTE support, all of them bought in mid-2015.

2.4 RRC-EIA0, no integrity protection on the RRC signaling

This vulnerability is a little more subtle than the one related to the global EIA0 mode (at both NAS and RRC levels) we found previously: there exists a case where a terminal enforces the integrity protection of the NAS signaling, but not of the RRC signaling. For this reason, an attacker with a fake LTE base station cannot establish a complete NAS signaling connection with the terminal, but can still eavesedrop on and modify user trafic.

When an LTE modem requests service to reestablish the user data connection to the Internet after an idle period, the uplink NAS message

Service Request is always unciphered (but integrity protected), and no downlink NAS message is required before the reestablishment of the data radio bearer... An attacker has thus the possibility of setting up a user data interception by configuring a fake LTE cell which has the same TAC as a legitimate one; this is to avoid the trigger of a *Tracking Area Update* NAS exchange with the targeted terminal. If the target comes close enough to the attacker in idle mode, it will camp on its fake cell; when the terminal requests service, the fake cell just needs to setup a data radio bearer after selecting no encryption and no integrity protection at the RRC level.

From here, the attacker will access all the subscriber user data in clear until the terminal reselects another cell or triggers a NAS procedure toward the network (e.g. a periodic *Tracking Area Update* or a modification in an ESM bearer).

2.5 Bypassing the NAS security dispatcher

Another serious vulnerability was found in which a terminal still accepts unprotected NAS messages, even after the security has been activated at the NAS layer (after a *security mode control* NAS procedure happens successfully). There is a 4 bit bitmask in the first byte of the NAS protocol header which indicates if a NAS packet is security-protected and encrypted or not; i.e. if the NAS header actually contains the NAS EMM security header or not.

After the activation of the security, a terminal conforming to the LTE standard must not accept any unprotected NAS message, this means any NAS message without the NAS EMM security header must be dropped; however this was not the case for a terminal we worked with. It enables an attacker with a fake LTE base station to inject NAS signaling, including SMS and LTE Positioning Protocol messages (abbreviated LPP) to the targeted baseband.

2.6 Revealing the IMEI in clear

In the LTE specification, only IMSI and temporary identity (TMSI, which is actually renamed GUTI in the LTE standard) can be requested and responded in clear during an *Identity Request* NAS procedure.

In the request, there is an identity type field which is encoded in 4 bit, but for which only the 3 least significant bits are used to encode the identity type:

- 1 (0b0001): IMSI;
- 2 (0b0010): IMEI;
- 3 (0b0011): IMEI-SV;
- 4 (0b0100): TMSI.

In our case, requesting the identity in clear with type 2 or 3 did not work to get the IMEI, however setting the most significant bit to 1 (which does not correspond to any standardized value) did bypass the check in the modem of one of the handset we worked with. So, requesting an identity of type 10 (0b1010) enables an attacker with a fake LTE base station to get the IMEI in clear, and an identity type of 11 (0b1011) enables getting the IMEI-SV in clear.

2.7 Revealing NAS ESM parameters in clear

When a terminal initially attaches to an LTE network, the *Attach Request* NAS message sent by the terminal is always unciphered (but can be integrity protected). During the attachment process, the terminal has the possibility of requesting the establishment of a default EPS bearer to a given APN. The LTE specification defines an *ESM information* procedure that allows the terminal to send specific data session parameters (e.g. APN name requested) during the attach NAS procedure, but after the NAS security has been activated. The 3GPP specification hence mandates that the *ESM Information Request* and *ESM Information Response* NAS messages are only exchanged within a secured NAS connection.

We could however find a modem which accepts responding in clear to unprotected ESM information requests; this enables an attacker to gather data session parameters such as APN name from a targeted terminal.

2.8 Overwriting the TMSI with an arbitrary identity

Temporary Mobile Subscriber Identity is used, in the form of GUTI, within LTE for the same purpose as TMSI within 2G and 3G networks. The network assigns regularly a new temporary identity to a terminal within a secured NAS connection.

An LTE modem was found which accepts the TMSI in a corrupted form, allowing for a certain restrained memory overflow. By carefully crafting a *TMSI Reallocation Command* NAS message, it becomes possible to overwrite the TMSI of the buggy modem with another kind of identity, e.g. an arbitrary IMSI or IMEI ! In subsequent NAS procedures, the modem will make use of this arbitrary IMSI or IMEI instead of its current temporary identity to try to identity itself to the network.

This bug has no real security impact itself, as it is required to establish a secure NAS connection with the modem to trigger it; it still illustrates how broken the memory management for the various LTE parameters exchanged over-the-air can be.

2.9 Memory management bugs

Memory management issues may cause buffer overflows in the modem, possibly leading to corrupting its execution. This class of bugs can allow for taking control of the modem by specially crafting a payload against its memory layout. This was demonstrated by Ralph-Philipp Weinman at DeepSec 2010 and USENIX 2012 [10], and more recently by Nico Golde and Daniel Komaromy at PACSEC 2015 during the Mobile Pwn2Own competition [23].

During our work on recent LTE modems, we could also trigger memory corruptions in some modems, most often resulting in crashing the modem's RTOS and resetting the cellular connection. We have not yet investigated those modem crashes.

It was even possible to find a memory management issue in a modem, which triggered a memory corruption in the Android RIL daemon after crashing the baseband. Luckily for the *rild*, the corruption was caught by a *FORTIFY_SOURCE* macro enforced at build time.

2.10 Other strange bugs

Some non-security related bugs were also found during our testing. A couple modems did not correctly implement the LTE specification, for example in the way they handle error cases:

- When a signaling message has not the correct format, the specification says a *status* signaling message has to be sent to report the error, which is not always done.
- Some modems do not respect all the timers following certain error or reject cases, continuing to issue connection requests and / or sending signaling messages to the LTE network. This can be an issue for MNO, however for us, this eases the process of fuzzing the baseband signaling stack !
- In another case, a modem was found to detach from the network with a corrupted value of its IMSI, appending 6 null digits at the end of it. This happened when the terminal was entering airplane mode, after the LTE network responded to the *Attach Request* sent by the terminal

with an invalid *Attach Reject* or *Attach Accept* NAS signaling message (the modem was hence trying to detach without being attached...).

- Finally, we found a case where the Android process *com.android.phone* crashes after the network delivers an IPv4 address in the multicast domain to it.

2.11 Bonus: tales from testing WCDMA stacks

As explained in [9], we also developed a WCDMA / UMTS testbed back in 2013, and we continue to use it from time to time in order to run basic tests against 3G modem stacks. Following are two security bypasses against 3G security procedures that we found in two different kind of terminals.

3G mutual authentication bypass

According to the 3G security standard TS 33.102 [5], when a 3G terminal equipped with a USIM card connects to a 3G radio access network, a 3G authentication procedure must happen. If the 3G network issues an authentication request without the $AUTN$ parameter, hence a 2G authentication, the terminal must reject the request with the cause «2G authentication unacceptable». There is a fundamental difference between a 2G authentication which implies a unidirectional terminal to network authentication, and a 3G authentication which is mutual between the terminal and the network and makes interception of 3G communications much more complex.

In our case, we could find terminals which were accepting 2G authentication when connecting to a 3G network. Those terminals were hence more likely to be intercepted by a 3G fake base station, as soon as the attacker was able to gather a single 2G security context of the target.

This issue existed in certain terminals of a given manufacturer, but was not present in the code base of the baseband manufacturer. Moreover the terminal manufacturer never acknowledged the issue when we reported it to him. The terminals concerned were produced around 2009 - 2010 and are now considered completely obsolete, they were never patched however.

Cross-domain security context mix-up

When a 3G terminal connects to a 3G radio access network, in the first step, a single signaling radio bearer is setup and used to transport both

CS (for voice calls) and PS (for IP connectivity) signaling. CS domain and PS domain being two different and independant systems in the mobile core network, each runs an authentication procedure and a security mode control procedure for securing the transport of user-data independently over-the-air. The first security context from one of the two domains being established is used to secure the initial signaling radio bearer. Then both CS and PS security contexts are used to secure a data radio bearer for the CS domain and a second data radio bearer for the PS domain.

We could find a 2G/3G/LTE modem which accepts the setup of both CS and PS data radio bearers after only a single authentication and security mode control happened in the initial signaling radio bearer. It is hence leaving one of the two user-plane radio channel in clear, because no security context is available for it.

The security consequence is not very important luckily, as an attacker will still need a valid 3G authentication vector in order to complete the signaling protocol exchange within the initial signaling radio bearer. The only advantage for an attacker will be that he will require a single 3G authentication vector to setup both CS and PS domain connections of the 3G modem toward his 3G fake base station.

2.12 The ciphering indicator

In accordance with previous research, we were not surprised to see no implementation at all of the ciphering indicator, as standardized in TS 22.101 [3]. The principle described in the standard is to have an indicator on the graphical user interface, that shows an explicit warning in case the radio connection stays in clear. Unfortunately, no recent handset implement this indicator (actually no handset in at least the last 10 years), and this is really a pity !

3 LTE modems analysis

As seen in the previous section, it is possible to find vulnerabilities and bugs with security implications in current multi-mode baseband. In order to investigate further those bugs, but also to make the testing process more reliable and automated, it is important to be able to monitor the proper functioning of the baseband and to extract information out of it when required.

LTE technology started to spread in 2010, with Qualcomm providing baseband processors and chipsets for most of the smartphone manufacturers. Today, other chipset manufacturers have been able to deploy their technology within smartphones worldwide. While Android is currently the main OS for smartphones, being widely open to system and application developers, the way cellular communications are implemented in cellular modems is kept closed and undocumented. We will try to analyze and explore how some of them work in this following section.

3.1 Previous work

In the work presented at SSTIC 2014 on the security of mobile baseband: « Analyse sécurité des modems des terminaux mobiles » [9] (french), information on 2G and 3G modems was presented after a large introduction on the state of the art.

Since then, not many articles have been released on this topic. The two most notable presentations and articles on cellular modem security and innerworkings are:

- The article [22] reveals implementations issues in LTE modems that allow breaking the privacy and availability of the LTE radio connection;
- The article « Insecurity of Voice Solution VoLTE in LTE Mobile Networks » [14] provides accurate information on how Voice over LTE is implemented with regards to the Android OS and the baseband.

3.2 Qualcomm

Qualcomm is the main manufacturer of cellular multi-mode modems for smartphones. It provides an integrated Snapdragon chipset (MSM) which is powered by a multicore ARM processor plus Hexagon DSP cores, two of them being assigned to the cellular modem (hence called mDSP). In this configuration, the telephony application within the Android OS is working with the modem through shared memory areas. Qualcomm also provides its modem in an autonomous chipset (MDM), which can be found in iPhones, starting with version 4S.

In recent Android smartphones, the modem firmware is loaded by the kernel into the dedicated Hexagon DSP. The file */system/etc/-firmware/mba* (modem boot authenticator) is loaded first by the PIL (Peripheral Image Loader). The file */system/etc/firmware/modem*, the modem executable, is loaded afterwards. Both binaries are in the form of

ELF executables for the QDSPv5 architecture, loaded at a fixed memory offset in RAM (this offset is available in the *dmesg* logs).

The modem binary is about 60 MB and runs all the network layers (L1, MAC, RLC, RRC, NAS, . . .) of the multi-mode modem, supporting many frequency bands and technologies: GSM, GPRS, EDGE, WCDMA (FDD and TDD), HSPA, CDMA-2X, HRPD, LTE (FDD and TDD). It is based on a Qualcomm proprietary embedded RTOS (called BLAST or QuRT), which starts more than 200 tasks at startup. A way to disassemble it was already proposed in our article at SSTIC 2014 [9]. Further work on a disassembly software for the QDSP6v5 ISA was done by Thomas Cordier, to whom we are thankful, during his master thesis at ANSSI.

From the Android environment, only limited information about what happens in the modem is available through Android *main* and *radio* logs. Qualcomm's specific diagnostic peripheral */dev/diag* allows obtaining more detailed information, including the raw RRC and NAS signaling messages received from and sent to the LTE network; a specific application available in the Android environment, *diag_mdlog*, makes use of this peripheral to generate a modem log file on the flash memory. Alternatively, one can redirect the diag kernel peripheral over USB (for instance on Samsung phones, with the command *setprop sys.usb.config diag,acm,adb* or the command *setprop sys.usb.config diag,adb* on LG phones, provided the kernel has support for DIAG).

This allows connecting Qualcomm's QXDM software, running on a PC, through a virtual serial link over USB. In 2011, Guillaume Delugré presented at the 28th Chaos Communication Congress (28C3) his work on reversing the MSM6280, using the DIAG protocol to read / write the modem's memory [1] and injecting a debugger.

While the DIAG protocol itself is largely undocumented, the aforementioned presentation provides many details about the inner working of this protocol. Moreover, a number of datasheets has surfaced over the year. In particular, a large amount of documents about the MSM8960 modem became public at the beginning of 2013, containing in particular the ICD (Interface Control Document) for GSM, UMTS and LTE. We also refer the reader to Guillaume Delugré's qcombbdbg as well as ModemManager's libqcdm; both of these open-source projects contain an implementation of the DIAG protocol.

A simple example is provided in the listing 1, showing how we can retrieve the NAS layer messages by sending a logging config request.

```
struct cmd_log_conf_request_set_log_mask
{
  int cmd_code;  /* CMD_LOGGING_CONF_REQUEST (0x73) */
  int operation; /* CMD_LOGGING_CONF_REQUEST_OPERATION_SET_LOG_MASK
      (3) */
  int equip_id;  /* CMD_LOGGING_CONF_REQUEST_EQUIP_LTE (11) */
  int num_items; /* 521 */
};
```

Listing 1. Activation of NAS messages reporting through DIAG

The structure *cmd_log_conf_request_set_log_mask* is followed by the 521-bit long bitfield; this size varies with each equipment identifier. To enable OTA (over the air) messages, we set byte 0x27 to 0x04 (corresponding to bit 314). The modem replies with an ACK of the log mask, and then starts sending the messages.

An example of an exchange between a french MNO and a Galaxy S3 LTE (i9305) phone at the beginning of 2016 is provided in listing 2.

```
[LTE NAS EMM] Uplink  : EMM_MSG_TYPE_ATTACH_REQUEST
msg = 0741620b f602f810 800368cf 3aa02504 e060c040 00050201 d011d152
     ...
[LTE NAS EMM] Downlink: EMM_MSG_TYPE_IDENTITY_REQUEST
msg = 075501
[LTE NAS EMM] Uplink  : EMM_MSG_TYPE_IDENTITY_RESPONSE
msg = 07560829 80108320 35040100 00000000 00
[LTE NAS EMM] Downlink: EMM_MSG_TYPE_AUTHENTICATION_REQUEST
msg = 07520610 ebaa0baf bc858e51 bbe419cd 5733da10 72d08585 dc348000
     ...
[LTE NAS EMM] Uplink  : EMM_MSG_TYPE_AUTHENTICATION_RESPONSE
msg = 07530843 016d9f1f da38b700 00000000 00
[LTE NAS EMM] Downlink: EMM_MSG_TYPE_SECURITY_MODE_COMMAND
msg = 075d2206 05e060c0 4070c1
[LTE NAS EMM] Uplink  : EMM_MSG_TYPE_SECURITY_MODE_COMPLETE
msg = 075e2309 33251305 06468008 f1000000 000000
[LTE NAS EMM] Downlink: EMM_MSG_TYPE_ATTACH_REJECT
msg = 07441378 00070201 d11b3701 bb
[LTE NAS EMM] Uplink  : EMM_MSG_TYPE_EMM_STATUS
msg = 07606000 00000000 00
```

Listing 2. Example of a NAS message trace, ending-up with an attach reject

For this particular case, we can use libmich to determine the cause of the attach reject. We see in listing 3 that the network did not accept the APN provided by the UE.

```
>>> from libmich.formats.L3Mobile import *
>>> show(parse_L3("0744137800070201d11b3701bb".decode("hex")))
```

```
### [ATTACH_REJECT] ###
 <Security Header Type [SH] : '0 : No security'>
 <Protocol Discriminator [PD] : '7 : EPS mobility management
     messages'>
 <[Type] : '68 : Attach reject'>
 <[EMMCause] : '19 : ESM failure'>
 ### [ESMContainer] ###
  <[T] : 120>
  <[L] : 7>
  <[V] : <[PDN_CONNECTIVITY_REJECT]: EBT(EPS Bearer Type):0,
  PD(Protocol Discriminator):'2 : EPS session management messages',
      TI(Procedure Transaction ID):1,
  Type():'209 : PDN connectivity reject', ESMCause():'27 : Missing
      or unknown APN',
  T3396():<[T3396]: T():55, L():1, V():'\xbb'>>>
```

Listing 3. Attach reject NAS message analysis with libmich

Cellular capabilities

Thanks to our LTE testbed, it is possible to gather the cellular capabilities announced by all basebands we have been testing. Here are for example the radio and security capabilities reported by a Sony Xperia compact Z3 (model D5803, mid-2015) with a Snapdragon 801 SoC (baseband version 8974-AAAAANAZQ-00049-40):

- LTE encryption capability: EEA0, EEA1, EEA2;
- LTE integrity protection capability: EIA1, EIA2;
- UMTS encryption capability: UEA0, UEA1;
- UMTS integrity protection capability: UIA1;
- GPRS encryption capability: GEA1, GEA2, GEA3;
- GSM encryption capability: A5/1, A5/3;
- radio interface compliancy: 3GPP release 10;
- UE category: 4;
- LTE bands supported: 1, 2, 3, 4, 5, 7, 8, 13, 17, 20;
- UMTS FDD bands supported: I, II, IV, V, VIII;
- GSM bands supported: 850, 900, 1800, 1900;
- LCS, LPP support: unknown.

3.3 Mediatek

Mediatek has been manufacturing cellular modems for a long time, and has recently gained a large market share in the smartphone area, providing integrated chipsets including a multicore 64 bit ARM CPU, a Mali GPU, DSP for processing multimedia data and a complete set of radio subsystems:

Bluetooth, Wi-Fi, GPS and a multi-mode cellular modem. We can find MTK chipsets in many entry-to-middle-level smartphones.

From the technical brief of its recent MTK 6735 chipset that is possible to find on the Internet, we can see that the cellular modem is running on a dedicated ARMv7 MCU at 600 MHz together with a DSP for the lower radio layers, within the SoC.

On the Android side, the modem binaries are within the directory */system/etc/firmware/*, in the files *modem_1_lwg_n.img* (ARM executable) and *dsp_1_lwg_n.img* (DSP image). They are simply loaded into the baseband memory by a dedicated Android driver. The *dmesg* logs related to the modem are prefixed with the *ccci* or *MD32* string. The kernel logs show interesting information regarding the way the modem executable is loaded and mapped in memory. One extract of these logs is provided in listing 4. A *klogcat* binary also provides additional logs compared to the default kernel and Android ones.

```
[5665627e.2a73674a] {E7,E7} <3> ccci_mdinit(1)| looking for /etc/
    firmware/modem_1_lwg_n.img...
[5665627e.2a7e5efb] {E7,E7} <3> ccci_mdinit(1)| find modem_1_lwg_n.
    img
```

Listing 4. kernel logs related to the MTK modem reloading

Moreover, additional files are related to the modem operation:

- the two files *md32_d.bin* and *md32_p.bin* near the modem binaries seem to be binary configuration files used by the modem;
- the files in the directory */data/nvram/md/NVRAM/* seem to be configuration files used by the modem too (likely storing the IMEI and other non-volatile data);
- the two files in */system/etc/mddb/* provide additional debug information about the modem executable.

It is possible to obtain modem logs through the *adb -b radio* command. Mediatek also provides a framework to extract very verbose logs from the modem by using the *MTKLogger* and *mdlog* Android applications. Even if there is no application startup link in the Android environment, it is possible to launch it with the adb shell command: *am start -a android.intent.action.MAIN -n com.mediatek.mtklogger/.MainActivity*. This makes possible to produce *rawmux* files which seems to be unfortunately a Mediatek proprietary and undocumented file format.

Cellular capabilities

Here are some radio and security capabilities reported by a Sony E4g (model E2033) with a Mediatek MT6732 SoC (baseband version MOLY.LR9.W1423.MD.LWTG.MP.V13.P35-2.41.J.1.023), bought in France in mid-2015:

- LTE encryption capability: EEA0, EEA1, EEA2, EEA3; (the ZUC cryptographic algorithm is supported)
- LTE integrity protection capability: EIA0, EIA1, EIA2, EIA3;
- UMTS encryption capability: UEA0, UEA1;
- UMTS integrity protection capability: UIA1;
- GPRS encryption capability: GEA1, GEA2, GEA3;
- GSM encryption capability: A5/1, A5/3;
- radio interface compliancy: 3GPP release 9;
- UE category: 4;
- LTE bands supported: 1, 2, 3, 5, 7, 8, 20;
- UMTS FDD bands supported: I, II, V, VIII;
- GSM bands supported: 850, 900, 1800, 1900;
- LCS, LPP support: both.

We can see here that the null integrity-protection EIA0 is supported by the modem. This does not mean however that it is possible to use it in an unsecure way, as explained in section 2. In the 3GPP standard, the EIA0 mode is required to support emergency calls over LTE with a terminal which has no USIM card.

3.4 Samsung

Samsung has been manufacturing multi-mode modems for several years, but they have begun to deploy them widely only recently, integrating them into high-end smartphones (e.g. in Galaxy S6, S6 edge and Note 4). An advantage of high-end smartphones is that there is no need to dismantle them in order to know what is inside, because somebody already did it and documented it on the Internet; which is what the people at Chipworks did. In April 2015, they published a freely available teardown and hardware analysis of the Samsung Galaxy S6 [11].

From the teardown report, we can see that the modem, called *Shannon 333*, is an independant chip stacked with the flash memory chip, next to the Exynos application processor which is stacked with the DRAM chip. There is a single DRAM chip for both the baseband and application processors.

From Android, we remark that the modem image is actually within a dedicated flash partition *sda8* (symlinked by *RADIO*), and modem configuration information is contained within another flash partition *sda3* (symlinked by *EFS*). The EFS partition is mounted within Android, whereas the RADIO partition is read by the Android binary *cbd* in charge of launching the baseband processor. From the startup scripts, we can find the exact command which launches the modem, as shown in the listing 5.

```
service cpboot-daemon /sbin/cbd -d -t ss333 -b s -m l -P platform
    /15570000.ufs/by-name/RADIO
```

Listing 5. startup script extract on modem launch

The binary uses an SPI link (*-b* option) to push the modem image from the *RADIO* partition to the baseband processor, and then sets an LLI link (*-m* option) for the main communication between the Exynos and the Shannon processors. This lead us to discover the Low Latency Interface technology [16], which enables a point-to-point interconnection between two physical chips and a communication « as if a device attached to the remote chip is resident on the local chip ».

The kernel logs are quite verbose about the Android part responsible for communicating with the modem: all logs starting with *mif* relate to the modem interface. Android has a serie of devices to handle all operations with the modem, all prefixed with */dev/umts_**. The modem image has a header which describes five parts: TOC (the header itself actually), BOOT (the bootloader executable for the Shannon processor), MAIN (the main modem executable), NV (which will map to a file from the EFS partition) and OFFSET. MAIN is around 43 MB long, and seems to be scrambled in some way.

The listing 6 shows extracts of the startup sequence of the modem.

- around 0.5s: Linux devices */dev/umts_** for baseband control and */dev/rmnet** for user-data transfer are created;
- around 3.8s: the modem image is read;
- around 4.1s: the modem image is splitted;
- around 4.8s: the LLI link is setup and the Shannon modem processor is reseted;
- around 5.1s: the BOOT code of the modem is pushed over the SPI link;
- around 6.5s: the modem enters update mode to download the main executable;

- during more than 3s: the MAIN executable is sent over the LLI link;
- around 14s: the NV part is sent to the modem;
- around 15s: the modem enters normal operation;
- from here: serial communication between the modem processor and the Exynos processor is established.

```
< 6>[    0.587830]  [4:      swapper/0:    1] mif: create_io_device:
     umts_ipc0 created
<11>[    3.839331]  [4:            cbd: 2972] mif: cbd: main:
    partition path : /dev/block/platform/15570000.ufs/by-name/RADIO
<11>[    4.141512]  [5:            cbd: 2972] mif: cbd:
    prepare_boot_args: toc[0].name = TOC
< 6>[    4.862580]  [4:            cbd: 2972] mif: print_mc_state:
    ss333: ss333_on: MC state:OFFLINE on:1 reset:1 active:0 status:0
     cp_dump_int:0 reason:0
< 6>[    5.173092]  [2:            cbd: 2972] mif: spi_boot_ioctl:
    IOCTL_MODEM_XMIT_BOOT (size 11080)
< 6>[    6.563099]  [7:            cbd: 2972] mif: misc_ioctl:
    umts_boot0: IOCTL_MODEM_FW_UPDATE
< 3>[    6.579138]  [4:            cbd: 2972] mif: check_udl_space:
    lli: NOSPC in RAW_TXQ{qsize:2084864 in:145176 out:145628 space
    :451 count:2072}
< 3>[   10.204779]  [5:            cbd: 2972] mif: check_udl_space:
    lli: NOSPC in RAW_TXQ{qsize:2084864 in:332168 out:332600 space
    :431 count:2072}
< 3>[   15.051183]  [7:            cbd: 2972] mif:
    io_dev_modem_state_changed: ss333->state changed (BOOTING ->
    ONLINE)
< 6>[   16.454783] I[0:      swapper/0:    0] mif: LNK-RX(48): f8 eb
     1d 00 19 00 00 00 13 05 03 0f 00 0d 0a 41
< 6>[   16.454810] I[0:      swapper/0:    0] mif: LNK-RX(36): f8 eb
     0c 00 08 00 01 00 01 01 03 01
```

Listing 6. kernel logs extract related to the modem startup

One way to unscramble the MAIN part of the modem image would be to make sense of the boot code in BOOT. Fortunately, the Android environment in Galaxy smartphones provides debugging menus to enable a ramdump of the baseband. The diagnostic control menu **#9090#* allows setting the modem debugging level, and the sysdump menu **#9900#* allows making a ramdump of approximatively 120 MB to the flash memory.

The ramdump is prefixed with a 32 bytes signature and then contains ARM code and volatile memory of the running modem processor. Many references to strings suggest the modem code is mapped at address 0x40000000, this makes possible to disassemble properly the executable. The exact structure of the RTOS has not been retrieved, however by searching strings, it is possible to reconstruct the control flow for parts of cellular operations.

Cellular capabilities

Here are some radio and security capabilities reported by a Samsung Galaxy S6 (model SM-G920F) with an Exynos CPU and Shannon 333 baseband processor (baseband version G920FXXU1A0D9):

- LTE encryption capability: EEA0, EEA1, EEA2, EEA3; (the ZUC cryptographic algorithm is supported)
- LTE integrity protection capability: EIA0, EIA1, EIA2, EIA3;
- UMTS encryption capability: UEA0, UEA1;
- UMTS integrity protection capability: UIA1;
- GPRS encryption capability: GEA1, GEA2, GEA3;
- GSM encryption capability: A5/1, A5/3;
- radio interface compliancy: 3GPP release 10;
- UE category: 4, ue-Category-v1020: 6;
- LTE bands supported: 1, 2, 3, 4, 5, 7, 8, 12, 17, 18, 19, 20, 26;
- UMTS FDD bands supported: I, II, V, VIII;
- GSM bands supported: 850, 900, 1800, 1900;
- LCS, LPP support: none.

Again, we can see that the null integrity-protection EIA0 is supported by the modem, but it does not mean that it is possible to use it in an unsecure way.

3.5 Other manufacturers

Other baseband manufacturers exist on the market:

- Hisilicon, subsidiary of Huawei, manufactures Kirin ARM SoC equipped with their Balong multi-mode modem; they are present in many cellular USB dongles and Wi-Fi bridges, and some top-level Huawei smartphones.
- Intel manufactures XMM modems; their XMM 7360 baseband processor is a multi-mode modem, however they are only known to equip few tablets and smartphones at this time (often accompanying their Atom application processor, e.g. in Asus Zenfone);
- Spreadtrum manufactures an ARM SoC with a multi-mode modem, used by few brands distributed mainly in Asia and Africa (e.g. Micromax, Gygabyte);
- Gemalto, through their acquisition of Cinterion, provides modems mostly for machine-to-machine systems;

- Sequans, a french company, offers OFDM modems (WiMAX and LTE), mostly for IoT platforms and machine-to-machine comunications.

The competition is harsh between manufacturers, and some of the previous cellular modem manufacturers did exit the market since our study in 2014 [9]:

- Renesas mobile licensed the Nokia baseband stack in 2009, and finally got bought by Broadcom in late 2013;
- ST-Ericsson stopped and sold its Novathor activity to Broadcom in early 2014;
- however Broadcom itself decided to stop its cellular modem activity in June 2014;
- Icera, which was acquired by NVidia in 2011, should be sold off or shut down in the 2nd quarter of 2016.

4 LTE security evolution

4.1 Evolution of LTE implementation in basebands

Enforcing the security of modems

As we have seen during our study, LTE modem stacks (but also 3G stacks) suffer from various bugs, some of them having serious security impacts on the private information of subscribers, some even allowing the complete interception of user data. The cellular modem is one of the most exposed piece of software worldwide, consuming untrusted data transmitted by base stations from kilometers around, on a 24 hours, 7 days a week, basis.

For those reasons, we hope baseband developpers are continously and seriously verifying their software implementations. Even if the cellular standards evolve quite rapidly, the most exposed code responsible for running basic operations in 2G, 3G and LTE is not huge and does not evolve a lot; it could be tested and validated extensively in a realistic way, without impacting too much the beloved time-to-market approach of manufacturers. Some companies are better than others in verifying and patching their code base.

All the bugs we presented in section 2 have been reported to modem manufacturers; all of them have acknowledged the issues and worked to provide patched modem firmwares. It is then up to the device manufacturers and mobile network operators worldwide to ease the distribution of firmware updates.

Providing more transparency

In order to give more confidence to LTE subscribers worldwide, modems manufacturers but also more generally handsets manufacturers and mobile network operators should show better transparency.

For the cellular modems, it would be really helpful to define a normalized interface to get logs and statistics out of the modems: this would offer the possibility for subscribers to see if their terminals are behaving normally or not. Furthermore, this would finally enable the implementation of the « ciphering indicator » on the graphical interface of handsets, which has been standardized for more than 15 years in the 3GPP TS 22.101 [3]. And after all, it would be really helpful to engineers when testing modems like we did.

4.2 Evolution in the standards and official organisms

Several changes in the 3GPP standards would also help to increase greatly the security level of LTE and future generation of cellular technologies. For example:

- Adopting a new key derivation scheme during or following authentication, in order to introduce *Perfect Forward Secrecy* for radio connection encryption. This would mitigate the compromise of authentication keys, which is certainly not frequent, but has a huge impact when happening (in particular when an HLR / HSS is compromised, involving millions of authentication keys).
- Introducing a way for modems to authenticate the *Broadcast Control Channel* (abbreviated BCCH) would greatly help to prevent, or at least detect active attackers on the radio interface. One possible way to do this could be the introduction of asymmetric cryptography, by introducing a new System Information Block (abbreviated SIB) containing a digital signature of all other important SIB broadcasted by eNodeBs. ECDSA signatures could be 512 bits long and would easily fit into the LTE BCCH, produced by eNodeB thanks to an operator-specific private key regularly renewed (e.g. every week). 256 bits public keys would then be securely distributed to active terminals during their initial network attachment. This would allow terminals to authenticate SIB messages sent over BCCH for all eNodeBs of their home network, enabling them to detect and possibly decide to not move to a fake LTE base station.

However, please note the method outlined here is a rough draft; it would require considerable effort to distribute and manage the keys, and in all cases would need further work to become part of the 3GPP standards.

A complementary approach would be to independently evaluate the conformance of modems against the security standards, and to ensure that basic out-of-specification test cases cannot circumvent the security procedures standardized.

Today, the Global Certification Forum (GCF) has the responsability of certifying 3GPP and 3GPP2-compliant modems in terms of radio features, capabilities and performance. However, it seems that they do not take the cellular security procedures and cryptographic algorithms into account. Another recent effort is currently done by the 3GPP and the GSMA in order to setup a security assurance methodology and evaluation process for network elements. Unfortunately modems seem currently not in the scope of this ongoing work. At this time, it appears that no real initiative exists to evaluate the security of cellular modems in a systematic way, this would however greatly further the goal of having more secure mobile handsets and modems.

As a global consequence, this would help making cellular communications more trusted and provide mobile network subscribers with more confidence in cellular technologies.

5 Acronyms

- LTE: Long Term Evolution, for mobile networks
- EPC: Evolved Packet Core
- EPS: Evolved Packet System
- 3GPP: 3rd Generation Partnership Projects, responsible for mobile networks standards
- MCC: Mobile Country Code, 3 digits (208 for France)
- MNC: Mobile Network Code, 2 or 3 digits (e.g. 10 for SFR)
- IMEI: International Mobile Equipment Identity, 15 digits
- PDN: Packet Data Network
- ANFR: Agence Nationale des Fréquences
- MNO: Mobile Network Operator
- TAC: Tracking Area Code
- VoIP: Voice over IP
- MME: Mobility Management Entity
- SGW: Serving Gateway

- PGW: Packet Data Network Gateway
- HSS: Home Subscriber Server
- UE: User Equipment (= handset / terminal)
- RRC: Radio Resource Configuration
- NAS: Non-Access stratum Signaling
- DRB: Data Radio Bearer
- SRB: Signaling Radio Bearer
- PDCP: Packet Data Convergence Protocol
- PHY: Physical Layer
- MAC: Media Access Control
- RLC: Radio Link Control
- EMM: Evolved Mobility Management
- ESM: Evolved Session Management
- SMS: Short Message Service
- LPP: Location Positionning Protocol
- S1AP: S1 Application Protocol
- IMSI: International Mobile Subscriber Identity
- 3G-AKA: 3G Authentication and Key Agreement
- SMC: Security Mode Control
- RAND: random authentication challenge
- AUTN: network authentication token
- XRES: expected response
- AMF: Authentication Management Field
- HLR: Home Location Register
- EEA: EPS Encryption Algorithm
- EIA: EPS Integrity-protection Algorithm
- KSI: Key Set Identifier
- TMSI: Temporary Mobile Subscriber Identity
- GUTI: Globally Unique Temporary Identity
- IMEI-SV: IMEI Software Version
- APN: Access Point Name
- RIL: Radio Interface Layer
- SIB: System Information Block
- BCCH: Broadcast Control Channel

References

1. 28C3: Reverse-engineering a Qualcomm baseband . https://events.ccc.de/congress/2011/Fahrplan/events/4735.en.html.
2. 3GPP specifications . http://www.3gpp.org/specifications/specification-numbering.

3. 3GPP TS 22.101 . http://www.3gpp.org/DynaReport/22101.htm.
4. 3GPP TS 23.401 . http://www.3gpp.org/DynaReport/23401.htm.
5. 3GPP TS 33.102 . http://www.3gpp.org/DynaReport/33102.htm.
6. 3GPP TS 33.401 . http://www.3gpp.org/DynaReport/33401.htm.
7. 3GPP TS 36.300 . http://www.3gpp.org/DynaReport/36300.htm.
8. 3GPP TS 36.401 . http://www.3gpp.org/DynaReport/36401.htm.
9. Analyse sécurité des modems des terminaux mobiles . https://www.sstic.org/2014/presentation/Analyse_securite_modems_mobiles/.
10. Baseband Attacks: Remote Exploitation of Memory Corruptions in Cellular Protocol Stacks . https://www.usenix.org/system/files/conference/woot12/woot12-final24.pdf.
11. Basic product teardown report of the Galaxy S6 . https://www.chipworks.com/about-chipworks/overview/blog/inside-the-samsung-galaxy-s6.
12. cellular networks on Wikipedia . https://en.wikipedia.org/wiki/Cellular_network.
13. cellular telecommunications on radio-electronics . http://www.radio-electronics.com/info/cellulartelecomms/.
14. Insecurity of Voice Solution VoLTE in LTE Mobile Networks . http://peng.cse.ohio-state.edu/pubs/li15-ccs.pdf.
15. La sécurité dans les réseaux mobiles LTE . http://michau.benoit.free.fr/CAESAR-2011_LTEsec_pub.pdf.
16. Low Latency Interface Specification . http://mipi.org/specifications/low-latency-interface-specification.
17. Minimal LTE corenet in Python . https://github.com/mitshell/corenet.
18. Observatoire 2G, 3G, 4G . http://www.anfr.fr/fr/gestion-des-frequences-sites/lobservatoire-2g-3g-4g/actualites/actualite/actualites/plus-de-22-000-sites/
19. Open-source implementation of LTE eNodeB, core network and UE . https://gitlab.eurecom.fr/oai/openairinterface5g/.
20. Open-source implementation of the 3GPP LTE specifications . http://sourceforge.net/projects/openlte/.
21. open-source LTE library for SDR UE and eNodeB . https://github.com/srsLTE/.
22. Practical attacks against privacy and availability in 4G/LTE mobile communication systems . http://arxiv.org/pdf/1510.07563.pdf.
23. Samsung S6 calls open to man-in-the-middle base station snooping . http://www.theregister.co.uk/2015/11/12/mobile_pwn2own1/.
24. sharetechnote . http://www.sharetechnote.com/.
25. Small Tweaks do Not Help: Differential Power Analysis of MILENAGE Implementations in 3G/4G USIM Cards . https://www.blackhat.com/docs/us-15/materials/us-15-Yu-Cloning-3G-4G-SIM-Cards-With-A-PC-And-An-Oscilloscope-Lessons-Learned-In-Physical-Security-wp.pdf